A Social-Based Approach to Mobile Edge Computing

Dimitri Belli[1], Stefano Chessa[1,2]
[1]Dept. of Computer Science, University of Pisa, Italy
[2]CNR-ISTI, Pisa, Italy
Email: stefano.chessa@unipi.it, dimitri.belli@di.unipi.it

Luca Foschini[3] and Michele Girolami[4]
[3]Dipartimento di Informatica: Scienza e Ingegneria
Università di Bologna
[4]CNR-ISTI, Pisa, Italy
Email: luca.foschini@unibo.it,michele.girolami@isti.cnr.it

***Abstract*— Mobile Edge Computing (MEC) opens to the possibility of moving high-volumes of data from the cloud to locations where the information is actually accessed. In turn, the combination of MEC with the Mobile CrowdSensing approach further increases the performance of sensing platforms designed to collect data produced by the crowd through wearable devices. In this work, we envision a MEC architecture composed by mobile and fixed edges. Their goal is to optimize the share of contents among users by exploiting their mobility and sociality. We first present an algorithms to identify a suitable set of mobile edges and we show how such selection increases the performance of a content-sharing scenario. Our experiments are based on the ParticipAct dataset, which captures the mobility of about 170 users for 10 months. The experiments show that the number of requests that can be served mobile edges is similar to that of requests served by fixed edges, and then that mobile edges can be considered a viable (and low-cost) alternative to fixed edges.**

Keywords—Mobile Crowdsensing, Human-driven Edge Computing, Social Mobility

I. INTRODUCTION

Interconnectivity is the keyword of the new millennium. Cities are slowly but progressively changing their geopolitics on the basis of new connection opportunities. But these aside, we live in an era where every object of some usefulness, indistinguishably, may be connected to the Internet. In this reference scenario, smartphones and wearable devices in general play an important role. These tools, emancipated from the primeval functionalities of means of communication between individuals, are equipped with short range communication interfaces allowing the exchange of information with each other, even in absence of any physical network infrastructure. As technology advances, new generations of sensors are embedded in these devices. Magnetometers, gyroscopes, accelerometers and many other sensors embedded in all personal devices and capable of collecting information at the location in which they are placed give rise to a limitless amount of data gathering. The enormous number of devices pervading the environments in which we move makes possible to exploit the people roaming in a seamless way by amplifying the collection and exchange of information through Mobile Crowd Sensing (MCS) platforms. In parallel Mobile Edge Computing (MEC), a recent technology evolving the two-layer cloud device integrational model, has lightened the computational load entrusted to devices. The idea behind this study aims at developing an additional information exchange model characterized by the synergistic use of MCS technology and MEC architecture to ease data dissemination through and among devices, whether fixed or mobile. In our previous work we proposed a Human-driven Edge Computing (HEC) model [1] to ease the deployment of MEC platforms and improve their scalability, and enabling more powerful MCS applications to dynamize an otherwise static information exchange platform. To enhance the information interchange between the cloud and personal devices, HEC flanks to the fixed network edges (FMEC) of basic MEC models, temporary mobile edges (M^2EC) selected within nodes of the network in a non-arbitrary way. It is possible to identify a M^2EC entity as a middleware proxy dynamically activated in places where people tend to stay for a while. By doing so, the standard three-layer hierarchical MEC architecture is enhanced in its middleware part with the introduction of a new kind of edges acting as good as the fixed ones overcoming issues related to social and spatial coverage. In this context, the selection of M^2ECs among network nodes is critical, and should privilege nodes that have better chances of data exchange with other nodes.

In this paper we propose a new algorithm for the selection of M^2ECs. The algorithm exploits the knowledge of the MEC users ties and communities to select the users that are better connected to a large fraction of other users as possible (i.e. users that are central to the community of users of the MCS). This because the devices of the selected users will also have a higher chance of being connected and thus to exchange data with a larger fraction of devices in the MCS, thus facilitating the flow of data to the M^2EC and, in turn, to the cloud. We assess this "social-aware" selection strategy of M^2EC by testing it over a dataset obtained from the ParticipAct MCS platform, and we show that the M^2EC mobile edges can reach the same performance in terms of collected data of the FMEC. Considering that M^2EC mobile edges are chosen among the devices of participants, they do not require maintenance and they can also change frequently to avoid overburdening the single device, the use of M^2EC mobile edges is thus a viable and low-cost alternative to fixed edges in the design of MCS platforms.

II. RELATED WORK

Without claiming completeness, in the following we briefly report some related work to build the needed background and have a prospect of the current state of the art of MCS, MEC and its evolution toward more dynamic HEC models able to leverage human mobility and sociality features.

Starting from MCS, in recent years, the widespread diffusion of mobile and wearable device paved the way to MCS as a new paradigm for collecting and sharing data. Leveraging human mobility and the pervasiveness of devices equipped with both sensors and short-range communication interfaces, MCS has opened new horizons to participatory and opportunistic sensing [2]. Some good and exhaustive surveys about more consolidated aspects in MCS area are available and address more consolidated aspects such as mobile crowd sensing and computing, user recruitment, as well as the idea of anticipatory mobile computing and networking to study repetitive user patterns and behaviors [3, 4, 5]. Recent efforts are aiming at tackling some still open issues in the area. Focusing on energy consumption, users who take part into-the-wild MCS campaigns often complain about energy consumption of their devices' battery [6], and a feasible context-aware solution to the problem is provided in [7]. Another hot topic is data quality. In fact, due to the low level of confidentiality in data exchange and data collection, the massive involvement of MCS volunteers is another very debated problem and recently some different solutions are emerging [8,9]. Environmental coverage is another timely and important aspect characterizing any MCS campaign and possible solutions to overcome this issue are suggested in [10].

Moving to MEC related efforts, MEC introduces a middleware layer as intermediary between the cloud and mobile devices. MEC aims at lightening the computational load entrusted to devices and a high responsiveness at the edge of the network deploying virtualized cloud resources at base-stations, thus approaching the computational capability from the cloud towards devices [11]. The main benefit of the MEC intermediate level is the reduction of computational costs and, from a communications perspective, the decrease of network load. One of the main shortcomings of MEC is that installation and maintenance operations of network edges have not negligible costs [12]. To overcome that issue, we recently explored the possibility of leveraging new research trends at the crossroads of MCS and mobile social networking to facilitate the identification and impromptu formation of M^{2}EC (as a low-cost alternative to traditional FMEC) [1].

Making a step further along the same direction, we conclude this section reporting some seminal activities about the proposal of the new HEC architectural model. In this new emerging area, the MEC/fog literature has already produced some relevant modeling work and some seminal design/implementation results. Narrowing to efforts close to ours, as reported in [13], some first exploratory research activities have considered cooperation issues between edges and the core, but only a very few works concentrated on the opportunities of having cooperation between devices and the edges. Considering MCS as an application scenario, [14] and [15] propose to enhance the MCS process by leveraging intermediate MEC nodes, namely, FMECs, to boost data upload from mobile nodes to the infrastructure [14] and to provide more computing/storage capabilities closer to end mobile devices [15]. A very recent and interesting work, the closest to our HEC concept for what relates to enabling more collaboration between entities co-located at edges is [16]: it proposes not only to have the traditional "vertical" collaboration between devices, MEC, and cloud level, but also an "horizontal" collaboration between entities at the same level via ad-hoc communications; however, it neglects humans and social/mobility effects, namely, there is no idea to dynamically identify and impromptu form M^{2}ECs as in our novel HEC proposal.

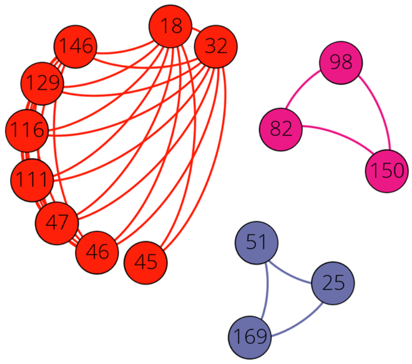

Fig. 1 Strong communities detected with TILES.

III. M^{2}EC SELECTION ALGORITHM

In our architecture, M^{2}EC are personal devices that, for a limited period, act as edges of the MEC architecture, to extend the coverage already offered by FMECs by providing the same services of the FMECs (or a subset of such services) to the other users' devices. Being a personal device of a user, a M^{2}EC is mobile and, for this reason, it covers a region of the spatio-temporal domain, according to its mobility. Consequently, it can provide its services to the devices that it encounters, i.e. that come within its low-range communication interface.

However, due their nature of being mobile devices carried by users, their opportunities of communication with other devices depend very much on the mobility of the users and on their relationships with other users of the MCS. For this reason, the selection of M^{2}EC should privilege devices of users that are

Algorithm 1 - M²EC detection

Let S be a strong community set $\{S_1, \ldots, S_n\}$ of a given CS
For all $S_i \in S$ do
  Let E_i be the ego network of S_i
  For all $u_j \in S_i$ do
    $B_j =$ betweenness_centrality (u_i, E_i)
  $M_i = \{\mathrm{argmax}(B_j)\}$ // M is the M²EC candidate for S_i
Sort $\{E_i\}$ // without loss of generality we assume that
    // $|E_i| \geq |E_j|$ iff $i<j$
Select as M2EC the first k nodes $M_1, \ldots, M_k$ such that:
- $| E_1 \cup \ldots \cup E_k | \geq \alpha$
- $| E_1 \cup \ldots \cup E_{k+1} | - | E_1 \cup \ldots \cup E_k | \leq \beta$

TABLE I.

Property	Value
α-value	70
Β-value	5
Dataset	ParticipAct
Algorithm	M2EC detection
Observation time	30 days
Location	Bologna (Italy) Lat 45° 27' 55.6'' Lon 9° 11' 11.4''
Number of participants	170
Observation period	30 days
Requests generated	$5x10^3$

more "social", i.e. that have frequent encounters with a larger number of other users. In other words, we aim at selecting M2EC to increase the social coverage of the MEC infrastructure, thus overcoming the limits of current MEC architectures due to their lack of human-based, mobile structures. To this purpose, we define al algorithm for M2EC selection that leverages social relationship among individuals. It is a fact that people move from one place to one another based on their habits and duties, and it is an equally consolidated fact that people forge social relationships on the basis of aggregational factors as gender, ethnicity, and so on [17]. Communities of our interest are those closely linked from these ties (Fig. 1 shows a graphical representation of the communities we detected with the dataset used - see Section IV-B). It is assumed that each community has within it a very limited number of nodes more cohesive than others, which form a strong community. These strong communities are the starting point for selecting M2ECs. Once a strong community of nodes have been identified, for each node we find the egocentric network, namely the set of nodes having ties with it (the union of the ego-networks of the nodes in a strong community forms an extended community). Then, we chose the representative node of the extended community between the nodes in the strong community based on the higher number of contacts with the extended community. Specifically, we use the betweenness centrality [18] to find such representative node of the extended community: that node is the candidate to become M2EC of that community. These steps for the selection of M2ECs are summarized in Algorithm 1.

In principle, the candidate for each strong community may be selected to become a M2EC, however, for efficiency reason, we select among these only those that contribute to the maximization of the social coverage of the system. To this purpose, we sort the ego networks according to their size, and we assign the role of M2EC to the candidates of the first k ego networks such that the union of the selected ego networks is above a given threshold α, and the contribution of the k+1 ego network to the union of the selected ego network is below a threshold β. The parameters α and β express a fraction of the entire population of the system and they are configured to avoid the inclusion as M2EC of nodes that do not allow a significant increment of the coverage of the M2EC.

The algorithm M2EC detection assumes that the communities and the strong communities are known. To this purpose, it keeps a social relation graph based on proximity factors among users. Specifically, the proximity among nodes is computed by taking measurements of distance through either GPS or devices' short-range communication interfaces as Wi-Fi in Direct mode or Bluetooth. To find the communities and the strong communities, it is possible to use any community detection algorithm. In our case, we used TILES [19], which is an algorithm for community discovery in dynamic social networks. In brief, TILES extracts overlapping communities tracking their evolution in time by a procedure based on pattern recognition. It uses a *domino effect* strategy re-computing nodes community membership whenever a new interaction occurs, and it does not impose fixed temporal thresholds for the partition of the network and the extraction of communities. An important feature is that TILES does not exclude from the computation overlapped communities, which means that each strong community node can belong to different strong communities. The latter represents the different spheres of the social world an individual belongs to.

IV. Experimental Settings

In the following, we describe the ParticipAct dataset used for our experiments, we present our experimental settings and we describe the results obtained in different scenarios.

A. The ParticipAct Dataset

The ParticipAct [20, 21] project provides a mobility dataset from the Emilia Romagna Region (Italy). The dataset covers approximately 15 months, from December 2013 to February 2015. To the purpose of this work, we restrict our observation up to October 2014 (10 months of observation period). The ParticipAct users use Android-based smartphones provisioned with the ParticipAct mobile app. The app is able to periodically compute the user position by exploiting the Google location APIs. The location is obtained by fusing together information coming from GPS, WiFi Hot Spot coordinates or cell phone base station. Each user's devices report its positions with a sampling period of 2.5 minutes. Users are mostly students of the University of Bologna, Italy. Therefore, their mobility strictly follows the mobility patterns of university students. Some of them commute daily from rural to urban areas, while other users roam most of the time inside the city center.

We extract from the ParticipAct dataset the co-location trace of the users, which does not track the global or the relative position of a device but, rather, tracks the start and end time of a contact between a device pair with a given time resolution. Therefore, the co-location trace can be considered as a time-evolving graph in which nodes are the user's devices and the edges report, at a given time, the connection between a pair. We consider that a device pair is in contact if the lie within 10 meters for at least a time slot of 2.5 minutes.

B. Simulation experiments

To test the efficiency of the selection algorithm presented in Section III, we exploit the co-location traces of the ParticipAct dataset, involving about 170 students. We restrict the location of the experiment to the city of Bologna for a period of 30 days. Specifically, our analysis covers from May 31 to June 30, 2014. We select such period in order to measure the performance of our architecture during a period in which students have

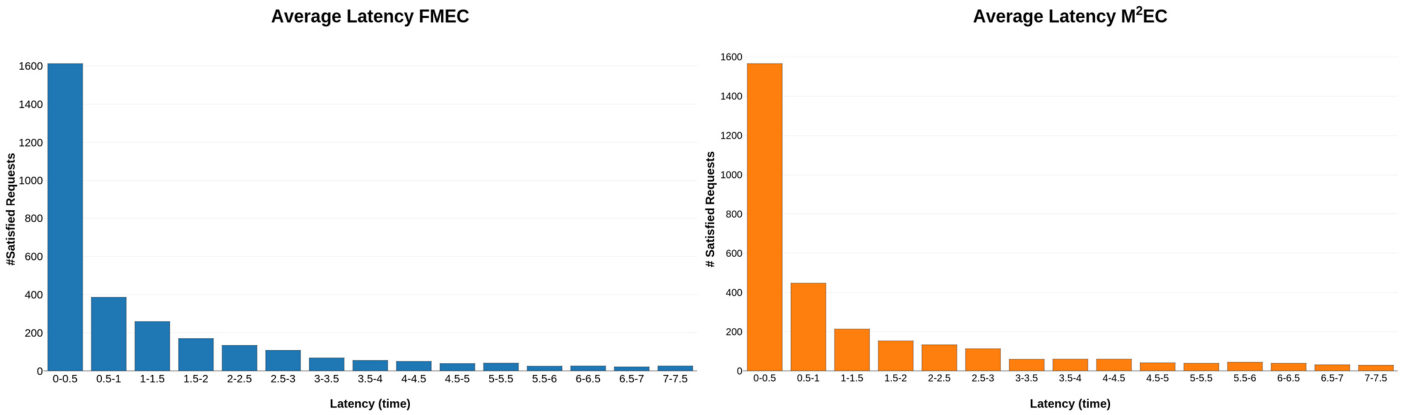

Fig. 1 FMEC and M^{2}EC latency.

meaningful social interactions with other students. The goal of our experiments is twofold. On one hand we want to select FMECs and M^{2}ECs with the algorithm presented in Section III, evaluating, in turn, the average latency of the requests satisfied by each kind of edge. On the other hand, we want to assess the performance of the M^{2}ECs in satisfying the requests raised by nodes belonging to the respective ego networks. Tab. 1 reports the configuration of our experiment.

We compare the results obtained in three different scenarios. Specifically, we consider a scenario in which only FMECs operate, a scenario in which only M^{2}ECs operate, and a scenario in which we combine together FMECs and M^{2}ECs. For each scenario, we consider two key metrics: latency and number of satisfied requests. Concerning the latency, it provides a measure of efficiency of our architecture in distributing contents of interest to nodes roaming in our scenario. We measure the following latencies:

- FMEC Latency: the average latency of requests satisfied by fixed edges only.
- M^{2}EC Latency: the average latency of requests satisfied by mobile edges only.
- FMEC & M^{2}EC Latency: the average latency of requests satisfied by both fixed and mobile edges.

For what concerns the number of requests satisfied, we provide an estimation of the effectiveness of our selection algorithm. We firstly compare the percentage of requests satisfied by FMECs only with the percentage of requests satisfied by M^{2}ECs only. Secondly, we evaluate the contribution of M^{2}ECs to our model. To this purpose, we measure the requests satisfied by a node belonging to the same community of a M^{2}EC or by other M^{2}ECs. Specifically, for each M^{2}EC we measure the following percentages:

- M^{2}EC in ego, which is the ratio between the number of requests satisfied by a specific M^{2}EC in its ego network and the total number of requests generated in such ego network.
- M^{2}EC out of ego, which is the ratio between the number of request satisfied by other M^{2}ECs generated in its ego network and the total number of requests generated in such ego network.

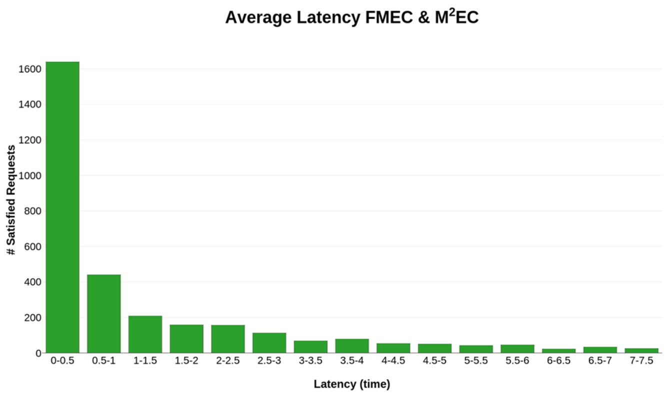

Fig. 2 Latency of FMEC combined with M^{2}EC.

- Missed ego requests, which is the ratio between the number of unanswered requests generated in the M^{2}EC ego network and the total number of requests generated in such ego network.

Our experiment works as follows. We first select a number of points of interest in which we deploy the FMECs. We consider that the number of FMECs is generally limited (this is due to deployment requirements, installation issues and costs of maintenance). In our case, we consider 6 locations where to deploy our FMECs [1]. Concerning the deployment of the M^{2}EC, we run the selection algorithm defined in Section III. As previously described, our algorithm relies on two key information: the strong communities and the centrality measures of nodes in ParticipAct.

C. *Simulation results*

We now present the evaluation of our solution. To this purpose, we consider those nodes in ParticipAct not acting as M^{2}EC and let them generate requests periodically (each request represent the need to access an edge, either fixed or mobile, to upload the data produced by the devices within a MCS campaign). Each request represents a specific resource needed by a node, such as multimedia contents, textual data or sensing information to collect. Moreover, each of the requests generated is represented as a pair: <*node ID, timestamp*>. We

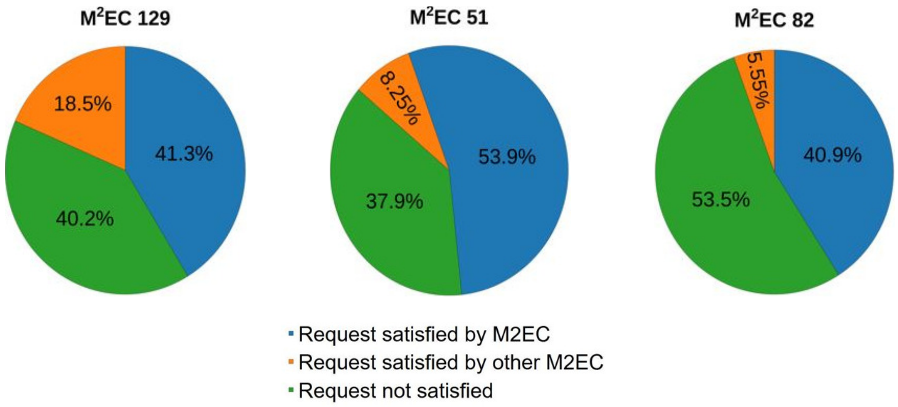

Fig. 3 Performance of M^{2}ECs

restrict the time of generation of the requests up to at three quarters of the total simulation time so that to give the time to all requests of being satisfied by the end of the simulation. This because the observation period is one month long, and the maximum limit for satisfying a request is one quarter of a month. As a result, each of the request generated has an equal possibility to be satisfied. For each of the requests, we measure its latency computed as: $t' - time$, where $t' \geq time$ is the time when the node generating the request encounters, for first time: a FMEC (FMEC Latency) or a M^{2}EC (M^{2}EC Latency) or any of the two previous (FM Latency). While the *time* variable is the current timestamp. The results concerning the latency metric are reported in fig. 1 and 2. From the two figures, it is clear that we obtain almost a similar trend, with long tails extending to the right and the majority of satisfied request concentrated in the first interval of time (0 – 0,5). A first general consideration is the following: although the number of FMEC acting in the simulation environment was double that of M^{2}EC, the latter managed a roughly equal performance of the former in terms of satisfied requests.

Fig. 2 shows the average latency obtained from the combination of FMEC and M^{2}EC. It is seen that the latency has a negligible reduction. Such behavior is expected because the number of FMECs is higher than that of M^{2}EC and their position is strategic in the territory. Therefore, FMECs generally serve the majority of nodes requests. All the histograms show a similar distribution, in both single and joint tests. Specifically, the majority of the requests raised by nodes in ParticipAct are satisfied in a relatively short time, by reporting an average latency of 1.5 days up to 7.5 days, time limit by which the raised request can be satisfied by fixed and mobile edges. Once this limit exceeds, a request is satisfied directly by the interaction of the nodes with the cloud without any communication with FMEC or M^{2}ECs.

Concerning the number of requests satisfied by FMECs and M^{2}ECs, we observe that the FMECs answer to a higher number of requests with respect to the M^{2}ECs. On average, the number of requests satisfied by fixed edges is 73.5%, whereas the the requests satisfied by mobile edge is 26.5%. We further investigate such percentages by running additional tests designed by varying the position of FMECs. We observe that , with specific settings, the percentage of the requests satisfied by M^{2}ECs can increase up to 40.4%.

We also study the performance of the M^{2}EC selected. In particular, we measure the percentage of requests generated from three sources: (1) generated by nodes belonging to M^{2}EC's ego networks, (2) generated by other M^{2}EC and finally (3) generated by the ego networks nodes, but satisfied neither by their M^{2}EC nor by others edges. We report in Fig. 3 the results for each of the 3 M^{2}EC selected. The results show that a M^{2}EC only (M^{2}EC 82 in Fig. 3) presents a number of un-answered requests barely higher than the number of satisfied ones. Whereas, all cases show a percentage of requests raised by an ego networks and satisfied by its own M^{2}EC much higher of the percentage of requests satisfied by other M^{2}EC. Out of a total of 5000 requests generated, over 3100, in several single and joined tests have been satisfied, with an average percentage of satisfied request of 62%.

From our previous experiments, we took the following observations. Firstly, our M²EC detection algorithm cannot operate without a prior assessment of the user's mobility. More specifically, the features of the human mobility can only be disclosed on a long-lasting data collection campaign. Datasets spanning for few weeks are not representative of dynamics of the human mobility, therefore they are useless for the selection of both FMEC and M²EC. Secondly, our mobile edge architecture relies on two key features of the human mobility: the spatial and the social coverage. Concerning the spatial coverage, we first analyze where to deploy FMECs by detecting those locations of ParticipAct crowded during the 24h. To this purpose, in [1] we describe a methodology for revealing points of interests by exploiting a spatial clustering algorithm. On the other hand, we also consider the social coverage, namely the capability of detecting *robust* communities of users in ParticipAct. To this purpose, we adopt the TILES algorithm as reported in Section III. The combination of spatial and social coverage allows to identify different kinds of edge nodes (FMEC and M²EC) acting as a bridge between the users, willing to access information anytime anywhere, and the cloud. A final consideration is related to the possibility of periodically electing nodes serving as M²ECs. Our results show the performance with a pre-defined number of M²ECs. However, we consider useful to study such performance with M²EC elected periodically, for example as soon as we detect significant deviations in the routine of human mobility. This is the case of crowded events such as sport matches, political meetings or anomalies in the regular traffic conditions. In these situations, the M^2EC detection algorithm presented in Section 3 could be re-executed so that to reveal new nodes that potentially can be elected as M^2ECs. We consider such last scenario as our next objective.

V. Discussions and conclusions

In this work we have investigated the potentials of using a Mobile Edge Computing (MEC) architecture to support mobile crowdsensing platforms. The advantage of using such an architecture lies in the fact that it eases the data dissemination through and among devices, whether fixed or mobile, since all communications of data acquired by the MCS platform may be collected by the edge nodes from the personal devices of the users by means of short-range communication links, thus increasing the scalability of the architecture and with no costs and a lower use of the users devices' resources as compared to broadband communication links. However, the introduction of MEC architectures pose new challenges related to the identification of the physical places where to install fixed edges, to the adaptability of the architecture to the changing user's mobility, and to maintenance of the fixed edges.

For these reasons we suggest to integrate the conventional architecture of MEC with mobile edges (M^2EC), which are some selected personal devices that, for a limited period, are requested to act as edges with respect to the other personal devices involved in the MCS platform. The advantage of introducing such mobile edges is two-fold: on the one hand, due to the mobility of their users they can opportunistically meet other devices of the MCS platform even in places where there are no fixed edges (and thus provide the services of the MEC in a more dynamic and adaptive way). On the other hand, they do not add extra cost to the architecture since they do not require installation and maintenance, but rather they can contribute to reduce the need for fixed edges, thus reducing the costs of the platform. However, due to their nature of being users' devices, they need to be selected based on the social characteristics of their users. In particular, their selection should privilege devices of users that are central to their communities and that thus have more chances of communication with other, non-M^2EC devices. To this purpose, we proposed an algorithm for the M^2EC selection that is based on the identification of communities of users. The results obtained by simulation over a mobility dataset of a real MCS platform show clearly that M^2EC can well integrate a conventional MEC platform and, possibly, even replace fixed edges, thus reducing the costs of the overall MCS platform.

Based on these results, we believe that M^2EC-based MAC architectures open new and interesting perspective for the future MCS platforms, but they also opens new research challenges concerning all aspects of the architecture, from the balance between fixed and mobile edges, to the way in which M^2EC interact with the rest of the platform and with fixed edges, to dynamic strategies for the selection of the M^2EC so that the burden of acting as an edge is better shared among the users and possibly many others. Currently, we are focusing our research efforts on the tuning of our selection algorithm to find the best mix of fixed and mobile edges.

References

[1] P. Bellavista, S. Chessa, L. Foschini, L. Gioia, and M. Girolami, "Human-enabled Edge Computing : Exploiting the Crowd as a Dynamic Extension of Mobile Edge Computing", IEEE Comm. Mag., vol. 56, no.1, pp. 149–155, 2018.

[2] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges", IEEE Comm. Mag., vol. 49, no. 11, pp. 32-39, 2011

[3] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm", ACM Comp. Surv., vol. 48, no. 1, pp. 7:1–7:31, Aug. 2015.

[4] H. Gao, C. H. Liu, W. Wang, J. Zhao, Z. Song, X. Su, J. Crowcroft, and K. K. Leung, "A survey of incentive mechanisms for participatory sensing", IEEE Commu. Surv. Tut., vol. 17, no. 2, pp. 918–943, 2015.

[5] V. Pejovic and M. Musolesi, "Anticipatory mobile computing: A survey of the state of the art and research challenges," ACM Computing Survey, vol. 47, no. 3, pp. 47:1–47:29, Apr. 2015.

[6] Y. Xiao, P. Simoens, P. Pillai, K. Ha, and M. Satyanarayanan, "Lowering the Barriers to Large-scale Mobile Crowdsensing", Proc. of the 14th Workshop on Mobile Computing Systems and Applications, 2013.

[7] X. Hu, X. Li, E. Ngai, V. Leung, and P. Kruchten, P., "Multidimensional context-aware social network architecture for mobile crowdsensing", IEEE Communications Mag., vol. 52, no. 6, pp. 78-87, 2014.

[8] S. Chessa, A. Corradi, L. Foschini, and M. Girolami, "Empowering mobile crowdsensing through social and ad hoc networking", IEEE Communications Magazine, vol 54, no. 7, pp. 108-114, 2016.

[9] P. Bellavista, A. Corradi, L. Foschini, and R. Ianniello, "Scalable and cost-effective assignment of mobile crowdsensing tasks based on profiling trends and prediction: The participact living lab experience", Sensors, vol. 15, no. 8, pp. 18613-18640, 2015.

[10] M. Girolami, S. Chessa, G. Adami, M. Dragone, and L. Foschini, "Sensing Interpolation Strategies for A Mobile Crowdsensing Platform", proc. IEEE Mobile Cloud, San Francisco, USA, pp. 102-108, 2017.

[11] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G", ETSI White Paper, vol. 11, no. 11, pp. 1-16, 2015.

[12] A. Ahmed and E. Ahmed, "A survey on mobile edge computing", Proc. 10th International Conference on Intelligent Systems and Control (ISCO), pp. 1-8, 2016.

[13] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications", IEEE Access, vol. 5, pp. 6757-6779, 2017.

[14] S. K. Datta, R. P. Ferreira da Costa, C. Bonnet and J. Härri, "oneM2M architecture based IoT framework for mobile crowd sensing in smart cities", Proc. of 2016 European Conference on Networks and Communications (EuCNC), pp. 168-173, 2016.

[15] K. M. S. Huq, S. Mumtaz, J. Rodriguez, P. Marques, B. Okyere and V. Frascolla, "Enhanced C-RAN Using D2D Network", IEEE Communications Mag., vol. 55, no. 3, pp. 100-107, 2017.

[16] T. X. Tran, A. Hajisami, P. Pandey and D. Pompili, "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges", IEEE Comm. Mag., vol. 55, no. 4, pp. 54-61, 2017.

[17] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks", Annual review of sociology, vol. 27, no.1, pp. 415-444, 2001.

[18] L. C. Freeman, "A set of measures of centrality based on betweenness", Sociometry vol. 40, no.1, pp.35-41, 1977.

[19] G. Rossetti, L. Pappalardo, D. Pedreschi, and F. Giannotti, "Tiles: an online algorithm for community discovery in dynamic social networks", Machine Learning, vol. 106, n. 8, pp. 1213-1241, 2017.

[20] G. Cardone, A. Cirri, A. Corradi, and L. Foschini, "The ParticipAct Mobile CrowdSensing Living Lab: The Testbed for Smart Cities", IEEE Commu. Mag., vol. 52, n. 10, pp. 78-85, 2014.

[21] S. Chessa, M. Girolami, L. Foschini, R. Ianniello, A. Corradi, and P. Bellavista, "Mobile crowd sensing management with the ParticipAct living lab," Pervasive and Mobile Comput., vol. 38, pp. 200-2014, 2017.